Cloud Imaging Model WG Minutes  
February 4, 2014

Meeting was called to order at approximately 1:00pm PT February 4, 2014.

Attendees

Andre Geertsema (Lexmark)  
G. Gupta (Oki Data)  
Smith Kennedy (HP)  
Daniel Manchala (Xerox)  
Ira McDonald (High North - call in)  
Joe Murdock (Sharp Labs America)  
Rainer Prosi (CIP4/Heidelberg)  
Rourke Randle (Toshiba)  
Jesse Sanchez (Intel)  
Tak Shiozaki (Epson)  
Mike Sweet (Apple - call in)  
Paul Tykodi (TCS - call in)  
Bill Wagner (TIC)  
Rick Yardumian (Canon)

Agenda Items

1. IP Policy and Minute Taker  
   a. Policy accepted with Mike taking the minutes
2. Review last minutes  
   a. ftp://ftp.pwg.org/pub/pwg/cloud/minutes/cloud-concall-
      minutes-20140120.pdf  
   b. Accepted as posted
3. Review of Cloud Imaging Model  
   b. Q: How does client discover cloud services?  
      - A: Two ways:
         - Client-side account credentials point to cloud system control
           service, and client uses ListAllServices operation to get a list
           of available services (similar to CUPS-Get-Printers and
           future IPP Get-Printers operation for IPP System Control
           Service spec)
         - LDAP/DNS-SD/etc. discovery protocols for public services
           (e.g. hotel managed printing services)
   c. Q: Does the IDS group deal with client/proxy association/registration
      issues?  
      - A: Yes, that will be one of the items addressed by IDS
   d. Q: Do we assume proxy can register with multiple cloud services?
- A: Yes

e. Q: Any kind of financial elements with proxy interface?
   - A: No, that is external to the model (i.e. there is a pre-existing business relationship that enables registration)

f. Section 4.1.2.1:
   - Line 727: "The proxy also periodically queries ..."
   - Line 728: "to check for waiting jobs" (drop "notification of")
   - Lines 729-734: Move "A failure to ..." after following paragraph, "failure to receive a query from the Proxy".

g. Section 4.1.2.2:
   - owner == Local Imaging System Owner

h. Q: What about conformance requirements?
   - A: Pretty loose for the model spec, binding specs will have the usual SHOULD, MUST, etc.
   - Historically, MFD Model had conformance requirements for operations and elements, but interoperability requirements tend to just be in the binding specs (e.g. IPP)

i. Section 4.1.4:
   - device -> equipment
   - May have one proxy talking to multiple Cloud Imaging Services of the same or different types.

j. Figure 4:
   - Show one Proxy talking to multiple Cloud Imaging Services

k. Q: Can multiple proxies be chained?
   - A: Conceptually yes. Fanout allows both direct (traditional Semantic Model/IPP interface) and indirect (the Proxy interface in the Cloud Imaging Model) usage, and this can be daisy-chained as needed.

l. Q: What about poll delays/responsiveness? Isn't directly talking to printer faster?
   - A: Bindings will likely provide long-running "get" operations - you "poll" to wait for notifications, response comes as soon as event is available
   - Talking directly *is* faster, however the purpose of this model is to enable imaging when the client is unable to create a direct connection to the service due to firewall or other network restrictions

m. Section 4.2.1.2:
   - Yes, we need to be able to target a device to conform to SM/IPP fan-out
   - May change name/terminology here to follow SM

n. Section 4.2.2, item 5:
   - Q: How to represent capabilities for multiple devices?
     - A: No way to report separate device capabilities as a single Local service
     - Solution: register multiple Local services, one per device
     - Solution: construct constraints that prevent combinations that are not supported (e.g. color, duplex, tabloid not
supported by any one device)
- Talk about implementation choice WRT intersection (only the common capabilities) vs. union (all capabilities) vs. separate local services

o. Section 4.2.2.3:
- Q: What about race conditions, e.g. two proxies fetching the same job?
  - A: While we don't talk about it here, in IPPSIX we use a "first proxy to fetch wins" approach, with the other proxy getting a "not fetchable" error
  - Add paragraph, "If the Job is no longer available to be fetched, an error is returned" (the model requires bindings to handle concurrency issues)

p. Section 4.2.2.9
- Don't want best effort for registration, if everything isn't OK then the response is an error with a list of elements that are not supported ("I can't do scan") or missing ("I need your geo-location")
- OK to ignore proxy info that the cloud doesn't care about (e.g. geo-location), do we need to report it?
  - IPP has successful-ok-ignored-or-substituted-attributes status code
  - What about reporting mobile printer geo-location in a moving car?
  - Might be useful to report attributes/elements that are not required (don't tell me about your geo-location)

q. Section 4.2.2.11:
- Should not allow proxy to deregister permanently, that is something you do through the cloud-specific interface, just like the initial setup prior to register
  - Q: Do we even need/want the operation?
  - A: Yes, because one proxy can register multiple systems

r. Section 4.2.2.13
- Response is just success or error (document not found, etc.)

s. Section 4.2.2.13/4.2.2.14:
- Q: Should we combine UpdateJobStatus and UpdateActiveJobs, to make a single UpdateJobs operation to update 1 to N jobs in one step?
  - A: No, see below
  - UpdateActiveJobs response, in all cases, returns list of job IDs and their updated states in the Cloud Imaging Service
  - New "invalid" job state for jobs that are not fetched or do not exist

T. Section 4.2.2.15:
- Eliminate GetCloudTerminatedJobs

u. GetServiceNotifications:
- JobTerminated: Add list of terminated jobs and their states

v. Figure 5:
- Add a second set of GetServiceNotifications requests after the
responses

w. Q: Can job states persist across re-registration?
   - A: Assume they can (not necessarily, but they could) and that the
     Proxy still does an UpdateActiveJobs to discover the fate of the old
     jobs
   - Cloud would move existing jobs to fetchable when the proxy doesn't
     include them in UpdateActiveJobs

x. Figure 6:
   - Drop GetCloudTerminatedJobs

y. Figure 9:
   - Drop GetCloudTerminatedJobs
   - Add GetJobElements operation
   - No longer just a JobTerminated flag - includes job ID

Next Steps / Open Actions

• Next Cloud conference call is February 17, 2013 at 3pm ET
• Action: Ron to find a Samsung editor (PENDING)